Network Telemetry on modern routers
Hello

I’m a software engineer with passion in computer networks and CTO / co-founder of FastNetMon LTD, London, UK
Ways to contact me

- linkedin.com/in/podintsov
- github.com/pavel-odintsov
- twitter.com/odintsov_pavel
- IRC, Libera Chat, pavel_odintsov
- pavel@fastnetmon.com
Disclaimer

None of the issues covered on this presentation are caused by vendor’s implementation. All of them are directly or indirectly caused by design of underlying protocols or standards.
Network telemetry on modern routers

- Netflow v5, v9
- IPFIX
- sFlow v5
- Port mirror
- Sampled port mirror (including GRE option)
- Raw headers over IPFIX or Netflow v9
## Protocol design: header

<table>
<thead>
<tr>
<th>Bytes</th>
<th>Contents</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>version</td>
<td>NetFlow export format version number</td>
</tr>
<tr>
<td>2-3</td>
<td>count</td>
<td>Number of flows exported in this packet (1-30)</td>
</tr>
<tr>
<td>4-7</td>
<td>SysUptime</td>
<td>Current time in milliseconds since the device booted</td>
</tr>
<tr>
<td>8-11, 12-15</td>
<td>unix_secs, unix_nsecs</td>
<td>Current count of seconds / nanosec since 1970</td>
</tr>
<tr>
<td>16-19</td>
<td>flow_sequence</td>
<td>Sequence counter of total flows seen</td>
</tr>
<tr>
<td>20</td>
<td>engine_type</td>
<td>Type of flow-switching engine</td>
</tr>
<tr>
<td>21</td>
<td>engine_id</td>
<td>Slot number of the flow-switching engine</td>
</tr>
<tr>
<td>22-23</td>
<td>sampling_interval</td>
<td>2 bits sampling mode and 14 bits sampling value</td>
</tr>
</tbody>
</table>
## Protocol design: flows, part 1

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-3</td>
<td>srcaddr</td>
</tr>
<tr>
<td>4-7</td>
<td>dstaddr</td>
</tr>
<tr>
<td>8-11</td>
<td>nexthop</td>
</tr>
<tr>
<td>12-13</td>
<td>input</td>
</tr>
<tr>
<td>14-15</td>
<td>output</td>
</tr>
<tr>
<td>16-19</td>
<td>dPkts</td>
</tr>
<tr>
<td>20-23</td>
<td>dOctets</td>
</tr>
<tr>
<td>24-27</td>
<td>First</td>
</tr>
<tr>
<td>28-31</td>
<td>Last</td>
</tr>
<tr>
<td>32-33</td>
<td>srcport</td>
</tr>
<tr>
<td>34-35</td>
<td>dstport</td>
</tr>
</tbody>
</table>
## Protocol design: flows, part 2

<table>
<thead>
<tr>
<th></th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>36</td>
<td>pad1</td>
<td>Unused (zero) bytes</td>
</tr>
<tr>
<td>37</td>
<td>tcp_flags</td>
<td>Cumulative OR of TCP flags</td>
</tr>
<tr>
<td>38</td>
<td>prot</td>
<td>IP protocol type (TCP = 6; UDP = 17)</td>
</tr>
<tr>
<td>39</td>
<td>tos</td>
<td>IP type of service (ToS)</td>
</tr>
<tr>
<td>40-41</td>
<td>src_as</td>
<td>ASN of the source</td>
</tr>
<tr>
<td>42-43</td>
<td>dst_as</td>
<td>ASN of the destination</td>
</tr>
<tr>
<td>44</td>
<td>src_mask</td>
<td>Source address prefix mask bits</td>
</tr>
<tr>
<td>45</td>
<td>dst_mask</td>
<td>Destination address prefix mask bits</td>
</tr>
<tr>
<td>46-47</td>
<td>pad2</td>
<td>Unused (zero) bytes</td>
</tr>
</tbody>
</table>
Benefits of Netflow v5

- Supported even by very old equipment
- Simple parser implementation due to static structures
- Simple sampling rate encoding (available in each packet)
Issues with Netflow v5

- Official standard does not exist
- Lack of IPv6 support
- Sampling cannot exceed 1:16384 due to 14bit
- Impossible to extend due to static structures
- Flow delays in range of 1–30 seconds before export
Netflow v9
Protocol design: template based

FlowSet 1 [id=0] (Data Template): 260
  FlowSet Id: Data Template (V9) (0)
  FlowSet Length: 100
  Template (Id = 260, Count = 23)
    Template Id: 260
    Field Count: 23
    Field (1/23): PKTS
    Field (2/23): BYTES
    Field (3/23): IP_SRC_ADDR
    Field (4/23): IP_DST_ADDR
    Field (5/23): INPUT_SNMP
    Field (6/23): OUTPUT_SNMP
    Field (7/23): LAST_SWITCHED
    Field (8/23): FIRST_SWITCHED
    Field (9/23): L4_SRC_PORT
    Field (10/23): L4_DST_PORT
    Field (11/23): SRC_AS
    Field (12/23): DST_AS
    Field (13/23): BGP_NEXT_HOP
    Field (14/23): SRC_MAK
    Field (15/23): DST_MAK
    Field (16/23): PROTOCOL
    Field (17/23): TCP_FLAGS
    Field (18/23): IP_TOS
    Field (19/23): DIRECTION
    Field (20/23): FORWARDING_STATUS
    Field (21/23): FLOW_SAMPLER_ID
    Field (22/23): ingressVRFD
    Field (23/23): egressVRFD

FlowSet 1 [id=0] (Data Template): 320
  FlowSet Id: Data Template (V9) (0)
  FlowSet Length: 100
  Template (Id = 320, Count = 23)
    Template Id: 320
    Field Count: 23
    Field (1/23): IP_SRC_ADDR
    Field (2/23): IP_DST_ADDR
    Field (3/23): IP_TOS
    Field (4/23): PROTOCOL
    Field (5/23): L4_SRC_PORT
    Field (6/23): L4_DST_PORT
    Field (7/23): ICMP_TYPE
    Field (8/23): INPUT_SNMP
    Field (9/23): SRC_VLAN
    Field (10/23): SRC_MAK
    Field (11/23): DST_MAK
    Field (12/23): SRC_AS
    Field (13/23): DST_AS
    Field (14/23): IP_NEXT_HOP
    Field (15/23): TCP_FLAGS
    Field (16/23): OUTPUT_SNMP
    Field (17/23): BYTES
    Field (18/23): PKTS
    Field (19/23): FIRST_SWITCHED
    Field (20/23): LAST_SWITCHED
    Field (21/23): IP_PROTOCOL_VERSION
    Field (22/23): BGP_NEXT_HOP
    Field (23/23): DIRECTION
Protocol design: sampling encoding

Cisco NetFlow/IPFIX
Version: 9
Count: 1
SysUptime: 1583525.3590000000 seconds
Timestamp: Mar 17, 2022 07:32:50.000000000 GMT
FlowSequence: 16488194
SourceId: 2081
FlowSet 1 [id=1] (Options Template): 257
  FlowSet Id: Options Template(V9) (1)
  FlowSet Length: 32
  Options Template (Id = 257) (Scope Count = 1; Data Count = 4)
    Template Id: 257
    Option Scope Length: 4
    Option Length: 16
    Field (1/1) [Scope]: System
    Field (1/4): FLOW_SAMPLER_ID
    Field (2/4): FLOW_SAMPLER_RANDOM_INTERVAL
    Field (3/4): FLOW_SAMPLER_MODE
    Field (4/4): SAMPLER_NAME
    Padding: 0000

Cisco NetFlow/IPFIX
Version: 9
Count: 1
SysUptime: 1860.000000000 seconds
Timestamp: May 24, 2016 17:02:27.000000000 BST
FlowSequence: 9 (expected 156)
SourceId: 525312
FlowSet 1 [id=1] (Options Template): 576
  FlowSet Id: Options Template(V9) (1)
  FlowSet Length: 36
  Options Template (Id = 576) (Scope Count = 1; Data Count = 5)
    Template Id: 576
    Option Scope Length: 4
    Option Length: 20
    Field (1/1) [Scope]: System
    Field (1/5): SAMPLING_INTERVAL
    Field (2/5): FLOW_ACTIVE_TIMEOUT
    Field (3/5): FLOW_INACTIVE_TIMEOUT
    Field (4/5): TOTAL_PKT_ENTRY
    Field (5/5): TOTAL_FLOWS_EXP
    Padding: 0000

[Expected Sequence Number: 156]
[Previous Frame in Sequence: 1]
Benefits of Netflow v9, part 1

- Supported by almost all vendors
- IPv6 support
- Can carry sampling rate in any range
- Well documented and most of the implementations are reasonably close to original implementation
Benefits of Netflow v9, part 2

- Offers almost unlimited extensibility
- Some fields are documented as part of IPFIX RFCs
Issues with Netflow v9, part 1

- Complicated data encoding for collector
- Sampling encoding is complicated and vendor specific
- Issues with flow duration encoding on some vendors
- Official standard does not exist
Issues with Netflow v9, part 2

- Tricky encoding for dropped by BGP Flow Spec traffic
- Lack of agreement between vendors about new fields
- Limited by subset of fields selected by vendor
- Flow export delay in range of 1-30 seconds
Protocol design: template based

<table>
<thead>
<tr>
<th>Template (Id = 256, Count = 29)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template Id: 256</td>
</tr>
<tr>
<td>Field Count: 29</td>
</tr>
<tr>
<td>Field (1/29): IP_SRC_ADDR</td>
</tr>
<tr>
<td>Field (2/29): IP_DST_ADDR</td>
</tr>
<tr>
<td>Field (3/29): IP_TOS</td>
</tr>
<tr>
<td>Field (4/29): PROTOCOL</td>
</tr>
<tr>
<td>Field (5/29): L4_SRC_PORT</td>
</tr>
<tr>
<td>Field (6/29): L4_DST_PORT</td>
</tr>
<tr>
<td>Field (7/29): ICMP_TYPE</td>
</tr>
<tr>
<td>Field (8/29): INPUT_SNMP</td>
</tr>
<tr>
<td>Field (9/29): SRC_VLAN</td>
</tr>
<tr>
<td>Field (10/29): SRC_MASK</td>
</tr>
<tr>
<td>Field (11/29): DST_MASK</td>
</tr>
<tr>
<td>Field (12/29): SRC_AS</td>
</tr>
<tr>
<td>Field (13/29): DST_AS</td>
</tr>
<tr>
<td>Field (14/29): IP_NEXT_HOP</td>
</tr>
<tr>
<td>Field (15/29): TCP_FLAGS</td>
</tr>
<tr>
<td>Field (16/29): OUTPUT_SNMP</td>
</tr>
<tr>
<td>Field (17/29): IP TTL MINIMUM</td>
</tr>
<tr>
<td>Field (18/29): IP TTL MAXIMUM</td>
</tr>
<tr>
<td>Field (19/29): flowEndReason</td>
</tr>
<tr>
<td>Field (20/29): IP_PROTOCOL_VERSION</td>
</tr>
<tr>
<td>Field (21/29): BGP_NEXT_HOP</td>
</tr>
<tr>
<td>Field (22/29): DIRECTION</td>
</tr>
<tr>
<td>Field (23/29): dot1qVlanId</td>
</tr>
<tr>
<td>Field (24/29): dot1qCustomerVlanId</td>
</tr>
<tr>
<td>Field (25/29): IPV4 ID</td>
</tr>
<tr>
<td>Field (26/29): BYTES</td>
</tr>
<tr>
<td>Field (27/29): PKTS</td>
</tr>
<tr>
<td>Field (28/29): flowStartMilliseconds</td>
</tr>
<tr>
<td>Field (29/29): flowEndMilliseconds</td>
</tr>
</tbody>
</table>
Protocol design: sampling encoding

Cisco NetFlow/IPFIX

Version: 10
Length: 72
- FlowSequence: 78350 (expected 279683213)
  Observation Domain Id: 524288
- Set 1 [id=3] (Options Template): 512
  FlowSet Id: Options Template (V10 [IPFIX]) (3)
  FlowSet Length: 56
- Options Template (Id = 512) (Scope Count = 1; Data Count = 10)
  Template Id: 512
  Total Field Count: 11
  Scope Field Count: 1
- Field (1/1) [Scope]: FLOW_EXPORTER
- Field (1/10): TOTAL_PKTS_EXP
- Field (2/10): TOTAL_FLOWS_EXP
- Field (3/10): systemInitTimeMilliseconds
- Field (4/10): exporterIPv4Address
- Field (5/10): exporterIPv6Address
- Field (6/10): SAMPLING_INTERVAL
- Field (7/10): FLOW_ACTIVE_TIMEOUT
- Field (8/10): FLOW_INACTIVE_TIMEOUT
- Field (9/10): collectorProtocolVersion
- Field (10/10): collectorTransportProtocol
Benefits of IPFIX

- Well documented RFC standard
- IPv6 support
- Unlimited flexibility
Issues of IPFIX

- Complicated encoding for collector
- Tricky encoding for dropped by BGP Flow Spec traffic (some vendors)
- Many vendors still do not support it
- Limited by subset of fields selected by vendor
sFlow
Protocol design: meta plus header

class __attribute__((__packed__)) sflow_sample_header_t {
    public:
    uint32_t sample_sequence_number = 0; // sample sequence number
    union __attribute__((__packed__)) {
        uint32_t source_id_with_id_type{ 0 }; // source id type + source id
        uint32_t source_id : 24, source_id_type : 8;
    }
    uint32_t sampling_rate{ 0 }; // sampling ratio
    uint32_t sample_pool{ 0 }; // number of sampled packets
    uint32_t drops_count{ 0 }; // number of drops due to hardware overload
    uint32_t input_port{ 0 }; // input port + 2 bits port type
    uint32_t output_port{ 0 }; // output port + 2 bits port type
    uint32_t number_of_flow_records{ 0 };
}
Benefits of sFlow v5

- Almost instant export (< 1 second)
- Provides access to packet header
- Simple sampling encoding
Issues with sFlow v5, part 1

- Sampling rate control is broken on almost all vendors
- Sampling rate selection process is tricky to grasp
- Traffic parsing is complicated and very hard to do in secure manner (IPv6 headers, MPLS, QnQ)
Issues with sFlow v5, part 2

- Lack of useful meta information (MPLS tags, VRF IDs, next hop)
- Long list of constraints and limitations from routers side (lack of LAG support for example)
Port mirror
Benefits of port mirror

- Complete access to all information in packet
- Supported by almost any router
Issues of port mirror

- Requires a lot of CPU time for collector to parse traffic
- Lack of meta information (ASN, VRF IDs, source and destinations ports)
- Requires spare ports on router
- Requires high performance network cards on collector
Sampled port mirror
Benefits of sampled port mirror

- Requires less port capacity
- Requires way less CPU on collector
- No need in high performance NICs
Issues of sampled port mirror

● Many vendors do not support it
● No way to get sampling rate, needs static setup
● Lack of meta information (ASN, VRF IDs, source and destinations ports)
● GRE requires MTU tuning to deliver 1500b+ packets
Payload via IPFIX or Netflow v9
IPFIX as transport for traffic headers

Cisco NetFlow/IPFIX
Version: 10
Length: 158
  FlowSequence: 7102
  Observation Domain Id: 16842752
- Set 1 [id=384] (1 flows)
  FlowSet Id: (Data) (384)
  FlowSet Length: 142
  [Template Frame: 109 (received after this frame)]
- Flow 1
  InputInt: 577
  OutputInt: 0
  Direction: Ingress (0)
  Data Link Frame Size: 1514

Data Link Frame Section: c8fe6a882418002cc83c85
IPFIX options as transport for sampling

- Cisco NetFlow/IPFIX
  - Version: 10
  - Length: 36
  - Timestamp: Mar 31, 2022 11:13:50.0000000000 BST
  - FlowSequence: 28436 (expected 0)
  - Observation Domain Id: 16842865
- Set 1 [id=3] (Options Template): 640
  - FlowSet Id: Options Template (V10 [IPFIX]) (3)
  - FlowSet Length: 20
- Options Template (Id = 640) (Scope Count = 1; Data Count = 1)
  - Template Id: 640
  - Total Field Count: 2
  - Scope Field Count: 1
    - Field (1/1) [Scope]: FLOW_EXPORTER
    - Field (1/1): SAMPLING_INTERVAL
  - Padding: 0000
Benefits of payload over IPFIX / Netflow

- That best and most capable protocol on market
- Almost instant traffic delivery
- Well defined format for sampling rate encoding
- Provides all information available in header
- Provides meta information (interface numbers, direction)
- Can be extended easily
Issues with payload over IPFIX / Netflow

- Only few vendors support it
- Extremely high complexity of integration for collector side
- Limited by set of fields provided by vendor
THANKS!

- pavel@fastnetmon.com
- linkedin.com/in/podintsov